

Nathaniel Serrano

COS 420

Program 7

4.9.2025

Lab Book

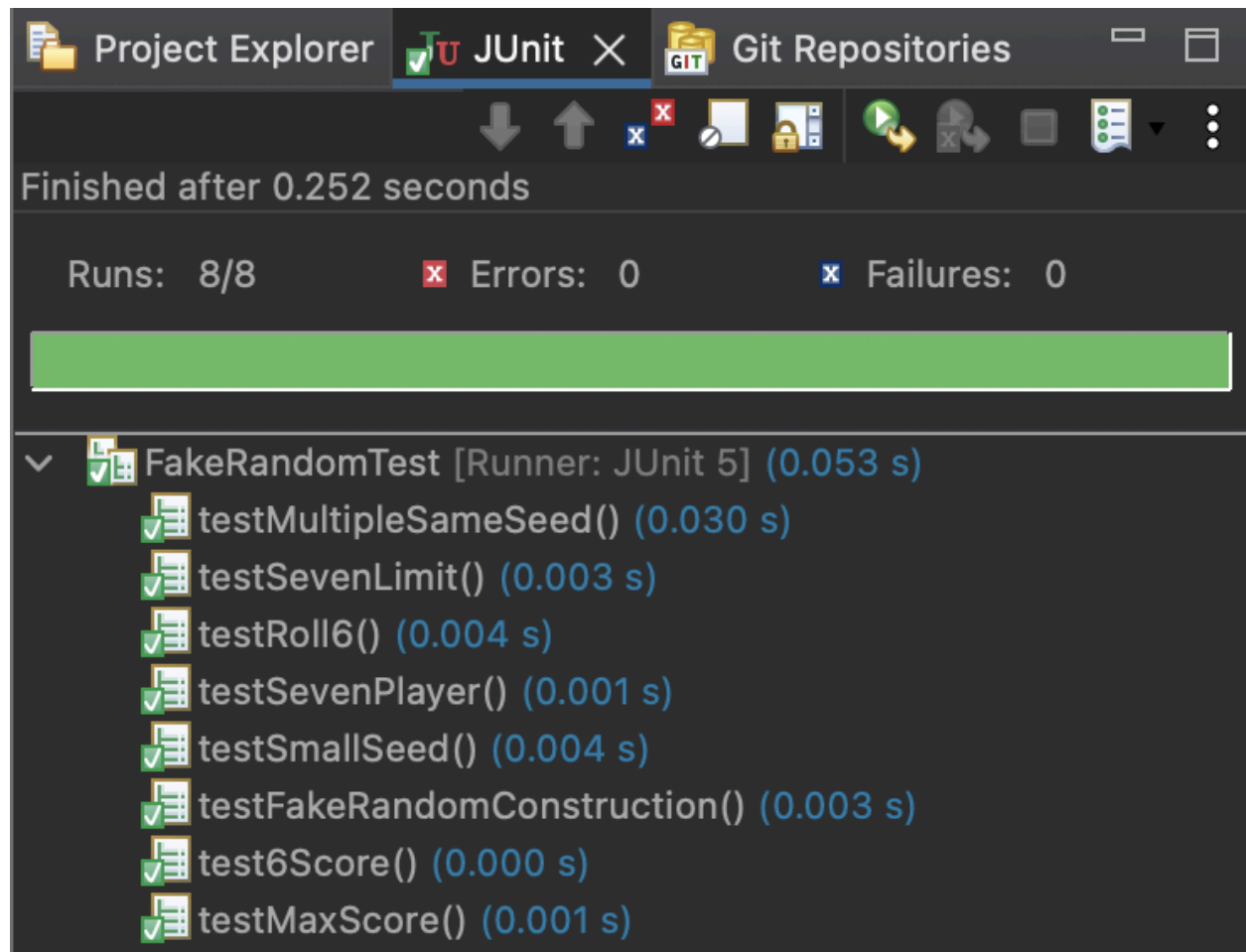
Time spent working with AI: April 8, 8:20 pm - 8:40 pm

Build a First Singleton

I began my prompting by ensuring it took place in the same session as the previous assignment, meaning Chat GPT 4o still had the full context of the Bulldog program and all of its classes. Next, I explained how I created the Referee.java class and provided the tool with a copy of the file. The prompt stated, “I have taken the part of the main method in Prog6.java that runs the actual game of Bulldog and made it into its own method in a new class called Referee.java. Rewrite this new class to be a singleton, following the object-oriented design pattern. Here is Referee.java below.” With that prompt, Chat GPT 4o rewrote the class into a singleton but made it have eager instantiation, which instantiates the Referee object as soon as the entire program is run. However, the tool offered to recreate it with lazy instantiation as an alternative and also offered to make the singleton thread-safe through double-checked locking to allow it to run properly in a multi-threaded environment. I decided to have the AI tool make the singleton utilize lazy instantiation instead since that was more accurate to the example we were shown in lecture. In terms of accuracy to the style shown in class, the AI tool did a great job. The constructor was marked private, with a Referee object named instance instantiated as null until getInstance() is called for the first time. Most of the time it took to complete this task came from preliminary work, like creating the Referee.java class and writing the prompt for Chat GPT 4o.

Once those were completed, implementing the AI tool's solution was simple, with no edits required outside of the request to utilize lazy instantiation. Overall, I was impressed with how easily it completed the task, and I was also quite impressed with how it offered different modifications to the singleton.

FakeRandom Test Screenshot



Reflection

As I mentioned previously when discussing the first singleton, I was impressed with Chat GPT 4o's performance with converting a class into a singleton. When writing the prompt, I made sure to include the words "object-oriented design pattern" when describing the singleton just to make sure that it understood what I was referring to. Because of its great performance with the

first singleton, I tasked Chat GPT 4o with the second singleton-related task found in Dice.java.

To complete this, I entered the prompt:

“Next, we're working in the Dice.java class. Change things so that the object of the Random class in Dice.java is a Singleton. Note that this is a bit different than making a class a Singleton, since you are only working with one attribute, but you should be able to figure out how to do it. When you are done, you will be able to have Dice of different sizes (6 and 2 would be the ones you would use - the former in all cases, and the latter for RandomPlayer) but all of those would share a Singleton Random object. Here is Dice.java in case you need it again: [code for Dice.java].”

The AI tool responded with:

“Perfect — you're right that this is a little different from making an entire class a Singleton. What you're really doing here is ensuring that all instances of Dice (regardless of size) share a single Random object across the entire application — essentially making the Random field a singleton resource.

The good news: you're already 90% of the way there! 

You just need to update the Random attribute to use lazy instantiation (like we did with Referee), but keep it static so it's shared across all Dice instances.”

Along with this explanation, the tool provided a modified Dice.java that was exactly what was asked with no issues. I quite like the response it gave, since it really does help the user understand what exactly is happening when converting the Random object into a singleton.

Small stuff like this allows the user to learn while still being productive, which I think is pretty valuable. In terms of performance, the usage of AI for these singletons definitely saved me time while also reaffirming what I knew about singletons. Of course, when it comes to learning from an LLM you have to be wary of hallucinations where the model will confidently state incorrect information. But as someone who already knew the basics of how singletons worked thanks to the class lectures, I was able to verify that the tool was correct in what it said while it affirmed what I knew, thus saving me time in completing the task while also serving as a form of education. Overall, a fun experiment with the AI tool.